

Simulation FDZ	Praktikum Software Entwicklung
LoggingAnalyse	26.03.2018

Logging Analyse

Kundenanforderung

Der Kunde verlangt eine Strukturierung des Logs durch Log-Level, diese können während dem laufendem Betrieb umgeschaltet werden. Außerdem soll der Log-Level in der Datei und der auf der Konsole unabhängig voneinander eingestellt werden können.

SLF4J Fassade

Das Projekt soll über einen längeren Zeitraum und von verschiedenen Entwicklern weitergepflegt werden, mit der Zeit können sich die Anforderungen an einen Logger verändern, deswegen wird die Fassade SLF4J¹ eingesetzt. SLF4J ermöglicht, ohne aufwändige Änderungen im Code, das Austauschen des Loggers und bietet einheitliche Methoden, um Logmeldungen zu erstellen.

Wahl des Loggers

Zur Auswahl stehen Log4J2², Java Util Logging (JUL³) und Logback⁴.

Vorteile der einzelnen Bibliotheken:

Log4J2	JUL	Logback
Gute Performance ⁵	Keine externe Bibliothek	Native Unterstützung von SLF4J Gute Performance

1 Offizielle Website: <https://www.slf4j.org/index.html>

2 Offizielle Website: <https://logging.apache.org/log4j/2.x/>

3 JavaDoc: <https://docs.oracle.com/javase/7/docs/api/java/util/logging/package-summary.html>

4 Offizielle Website: <https://logback.qos.ch/manual/introduction.html>

5 Performance Test: <https://www.sitepoint.com/which-java-logging-framework-has-the-best-performance/>

Simulation FDZ	Praktikum Software Entwicklung
LoggingAnalyse	26.03.2018

Nachteile der einzelnen Bibliotheken:

Log4J2	JUL	Logback
Externe Bibliothek	Schlechte Performance	Externe Bibliothek

Logback und Log4J2 sind beide klar vor JUL zu bevorzugen da sie eine deutlich bessere Performance liefern, die gerade für den Speedmode der Anwendung vorteilhaft ist. Zwischen Log4J2 und Logback fällt die Entscheidung schwer letztlich ist die native Unterstützung von SLF4J von Logback, der ausschlaggebende Aspekt um sich für Logback zu entscheiden.